Mars Odyssey Aerobraking: The First Step Towards Autonomous Aerobraking Operations

Mark A. Johnson, William H. Willcockson
Lockheed Martin Space Systems
P.O. Box 179
Denver, CO 80201-0179
mark.z.johnson@lmco.com
william.h.willcockson@lmco.com

Abstract— The 2001 Mars Odyssey spacecraft represents the third successful aerobraking mission. Aerobraking is a means of reducing the amount of propellant carried by an orbiting spacecraft. Achievement of the target science orbit is made through numerous light drag passes through a planet's upper atmosphere. Although there are significant mission advantages afforded by aerobraking such as a smaller propulsion system and increased scientific payload, they have typically been offset somewhat by increased operational risk. The risk is due to the long period of spacecraft operations (relative to the single event of orbit insertion) that must be performed in order to complete aerobraking. As this phase of the mission progresses, the ground operations team must continue to make important decisions under ever tightening time constraints. In an effort to simplify the tasks performed by ground personnel, a basic level of autonomous capability has been introduced on the Odyssey spacecraft that will lead the way towards reducing the risks and improving the efficiency of aerobraking.

The central part of this first level of autonomy is the Periapsis Time Estimator (PTE) software – an algorithm that resides in the on-board flight software and uses drag pass acceleration data to compute the error between the predicted and actual periapsis times for each orbit. When activated, the computed error is used to keep the execution of commands synchronized in time with what is really happening in the orbit. The current software is intended to operate in short orbits, when command sequences for multiple orbits have been loaded on-board in advance. These short period orbits, accounting for roughly one third of the total duration of the aerobraking phase yet well over half of the total number of drag passes, are the most critical and stressful for the operations team because it is difficult, and eventually impossible, to respond to the outcome of one drag pass before commanding begins for the next. Problems, such as loss of uplink/downlink, commanding errors, or spacecraft anomalies, are amplified due to the short periods of time available between drag passes, and can be catastrophic if they are not resolved in a timely manner. Even if the spacecraft and ground operations team are working nominally, atmospheric variability will cause timing errors that must be dealt with.

Originally, aerobraking operations for the Odyssey spacecraft were designed without the autonomous software in mind. Ground-based contingency functions such as the "Jack" procedure (a manual process that accomplishes the same task as PTE), and the on-board auto-popup capability – which autonomously performs a periapsis raise maneuver if the spacecraft enters a safing condition – were developed on prior missions and were available for use. Had the PTE software not proven reliable and useful, these functions would likely have been employed. The PTE software design and operability will be described, including presentation of flight results from the Odyssey aerobraking phase. An analysis of the software's strengths and weaknesses will also be presented. Finally, we will outline areas for improvement in the current autonomous software implementation, as well as additional capabilities that may be added in upcoming aerobraking missions, such as autonomous decision capability for flight corridor control and autonomous command sequence generation. The combination of all these elements could result in a spacecraft performing aerobraking with a reduced need for ground interaction, increasing robustness and efficiency.

[2] IEEEAC paper #1169, updated January 10, 2003

TABLE OF CONTENTS

1. INTRODUCTION

Nominal Odyssey Aerobraking

The Mars Odyssey Orbiter's mission is to perform Mars science observations from a 400 km circular polar mission orbit. In order to save on launch vehicle costs, the use of aerobraking was baselined to lighten the vehicle through greatly reduced MOI propellant. Aerobraking is the use of multiple low energy drag passes to reduce the orbital energy of a spacecraft over time. This technique had been used successfully on two prior planetary missions, that of Magellan at Venus [1] and Mars Global Surveyor (MGS) at Mars [2]. The aerobraking strategy utilizes propulsive capture into a large Mars parking orbit, followed by several hundred aerobraking drag passes to gradually circularize the vehicle over several months time. By flying at a sufficiently high altitude in the atmosphere, the heating of any drag pass is low enough to minimize thermal impact to spacecraft systems. The ground operations team tracks the vehicle and commands periodic correction burns to raise or lower the orbital periapsis to achieve desired drag levels.

For each drag pass, the spacecraft is first configured into a high-drag geometry (Figure 1) which is aerodynamically stable. Prior to reaching the atmosphere, the vehicle's –Y axis is aligned with the local velocity vector using onboard polynomials developed from ground navigation solutions. This presents the maximum drag area to the aeroflow and is close to the natural trim attitude of the spacecraft. The attitude control system is reconfigured to utilize thrusters and a rate-damping controller maintains stability during the drag pass itself. Once the drag pass is complete the solar array is unstowed and the high gain antenna is pointed to Earth for downlink of telemetry. The decay rate is periodically adjusted through the use of aerobrake maneuvers (ABM's) which are burns performed at apoapsis to adjust the periapsis altitude within the atmosphere. Due to the accuracies of long-range tracking and Orbit Determination (OD), periapsis control accuracies of 1.5 km have routinely been achieved.

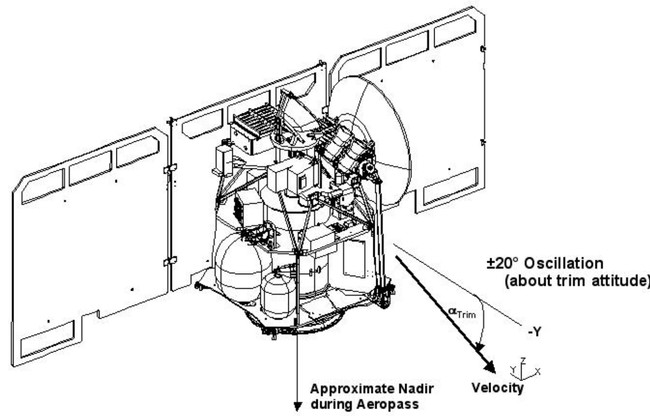

Figure 1 - 2001 Mars Odyssey in Aerobraking Configuration

One of the key environmental issues any aerobraking vehicle has to deal with is the variability of the high-altitude atmosphere. For Odyssey, maximum heating rates are monitored at each drag pass, and orbit periapse altitude is adjusted to minimize thermal impacts to the spacecraft. This requires that the spacecraft fly above 100 km altitude for most drag passes. This altitude is in the lower thermosphere where densities are very low and solar influences strong, resulting in large atmospheric variability. Typical orbit-to-orbit density variations at Mars have a standard deviation of 35%, as observed by MGS. In addition, the eruption of large Mars dust storms is a continual threat through much of the Martian year. These dust storms can raise the local densities by an order of magnitude in a few days time.

The consequence of this atmospheric variability is that spacecraft orbit predictions get stale fairly quickly, since the ground-predicted future orbits cannot anticipate the dynamic atmosphere. As the navigation solution gets stale the position of the vehicle on the predicted orbit becomes increasingly inaccurate. This causes chaos with spacecraft sequences, which must be built hours to days in advance of their execution. Thus, due to these orbit timing errors, the spacecraft events are executed at the wrong time – a serious problem if the vehicle encounters the atmosphere when expecting to be in vacuum. Timing errors can rapidly drive the spacecraft into "safe mode" as it gets confused – unexpected aerodynamic forces can be interpreted as controller or actuator failures, a very undesirable situation. Although the timing design of the onboard sequence includes buffers of 5 minutes on each side of the drag pass to help accommodate these prediction inaccuracies, the asymptotic nature of timing error growth can rapidly overwhelm the system.

Having dealt with these aerobraking timing problems, we began looking into autonomous aerobraking techniques as early as December 1994. It took a number of years , until February of 1999, for these concepts to coalesce into a set of workable requirements. From there an internally funded study of algorithms to mitigate onboard timing error correction was conducted. We concluded that the presence of highly dynamic aero forces, as detected by onboard accelerometers, could be feasibly used to correct the onboard timing by means of a fairly simple feedback loop after each drag pass.

robustness testing, system interactive testing, and various data failure modes.

2. ESTIMATION OF PERIAPSE TIME

Concept

The Periapsis Time Estimation (PTE) algorithm was developed as a means to shift the execution of an onboard command sequence in time based on calculations performed

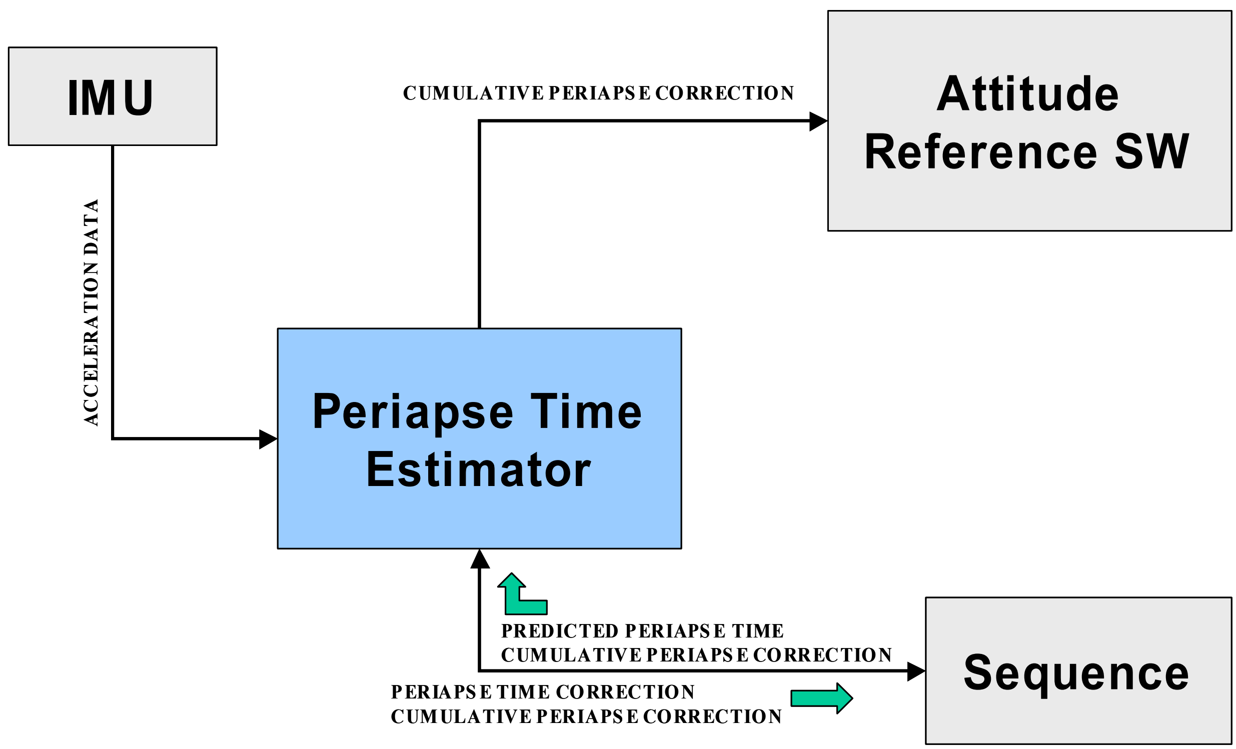

Figure 2 – Basic PTE Block Diagram

Started as an internal research and development (IR&D) project, the ultimate goal of this effort is to eventually achieve truly autonomous aerobraking where the ground operation's currently intensive efforts are greatly reduced. We envisioned this to occur in three stages. Phase I was the detection and correction of onboard sequences using drag detection. Phase II added the capability to perform limited orbital changes using autonomous correction burns. Phase III expanded the magnitude of corrective burn capability as well as adding onboard generation of aerobraking sequences using pre-defined rules.

The IR&D design effort looked attractive to LMA's Odyssey program management and was thus accepted in February 2000 as a development item for the aerobraking orbiter. The software module, which performed the timing corrections was called Periapsis Timing Estimator or PTE. The principal effort associated with flight certification was

on accelerometer data collected during a drag pass. It is part of the on-board flight software, and once configured and activated, it operates autonomously. It allows sequence timing adjustments to be made if the ground operations team were unable, either due to communication problems or due to time constraints. PTE, as implemented on the Mars Odyssey spacecraft, is a reactive process; it shifts the upcoming orbit based solely on the error detected during the last drag pass. Unlike some of the ground-based tools available to the operations team, it has no predictive capabilities to account for differences between the observed delta-V magnitude and the predicted delta-V magnitude achieved from atmospheric drag. PTE only compares the observed periapsis time to the predicted periapsis time. As a result, the PTE timing correction tends to lag by one orbit. This makes the PTE software more effective in tracking and correcting timing error trends, rather than statistical fluctuations.

The PTE software computes its estimate of timing error by examining the accelerometer data throughout the drag pass. It uses a centroiding scheme to find the "time-center" of the acceleration curve (see Figure 3). The algorithm is started at the beginning of the drag pass command sequence and stopped at the end of the sequence to guarantee that the entire drag pass is captured. During operation, filtered acceleration data is collected from the IMU processing software and weighted with time (see Figure 2). Data is rejected if the acceleration levels are below a minimum threshold (to account for bias and noise in the accelerometers). Computation of the time of periapse passage occurs when the algorithm is commanded to stop, but only if sufficient data has been collected. PTE's estimate of periapsis time is compared to the navigation prediction of periapsis time, and the error is computed as the difference between the two. Corrections are made for filter lag and planetary oblateness effects. The timing error is then subject to several thresholds and "goodness" criteria. If configured to act on a command sequence, the resultant timing error is used to shift the execution of all subsequent drag passes within a sequence. Failure to meet all of the goodness criteria will result in no shift, regardless of the computed value.

Development and Testing

Since the Periapse Time Estimator was flown as a demonstration on Mars Odyssey, its ability to affect the on-board sequence was constrained in many ways. Some of these limitations were hard-coded in the software itself, such as ensuring that the spacecraft was operating normally before performing any calculations. The majority took the form of limits and thresholds that were specified in an uploaded data file. The need for these limits was identified during the algorithm development and integration process. While these constraints would help prevent the software from issuing unwanted, or erroneous time adjustments, their functionality and impact on other flight software had to be fully tested first.

Operation of PTE algorithm is governed by configurable parameters. The algorithm can be configured to operate in a "watch mode" (computing but not executing corrections), only return a correction on certain drag passes, or even return corrections only within a definable magnitude range. The "raw" calculated correction is available via telemetry to gauge what the algorithm would have done if permitted.

The centroiding process used by the algorithm assumes only one drag pass of data is collected. A test is performed on the predicted periapse time supplied by the onboard sequence and the amount of data collected to help make sure this assumption is valid. Additionally, the algorithm will be stopped at the end of every pass, making it extremely unlikely that multiple drag passes would be encountered in a single data collection window.

The algorithm is intended to operate only under "nominal" spacecraft operating conditions during the aerobraking mission phase. Operation during contingency situations is not desired, due in part to the significant increase in complexity and required testing this would mean.

To certify the algorithm for flight operation, three different categories of tests were run – general performance tests, fault response evaluation, and flight software interaction tests.

To facilitate rapid performance analysis of the PTE software in a representative set of operating conditions, a majority of the drag acceleration data from Mars Global Surveyor (MGS) aerobraking operations (both Phase 1 and Phase 2) was used as a test data set. The set was visually examined to remove passes that were missing data (telemetry dropouts) or were otherwise corrupted. The remaining profiles, combined with a file of spacecraft sequence periapse times resulted in a set of 450 cases to be tested. The majority of the performance-driven testing was performed utilizing an engineering development version of the PTE algorithm, which was created as part of a MATLAB-based simulation. It was developed with performance testing of the PTE software in mind, and as such did not have to accurately simulate the remainder of the spacecraft flight software environment. As the MGS drag profiles were comprised of actual accelerometer output, the data was processed by a Simulink model of the Odyssey IMU and IMU Processing software before being fed to the PTE algorithm.

In the interest of speed and simplicity, the MGS data set was processed using a single set of PTE configuration parameters. The consequence of this approach was that the oblateness of Mars was not properly accounted for during the running of this test, but was accounted for in the post-processing of the data. The correction lookup depends on both the apoapse (given a relatively stable periapse altitude) and the argument of periapse of the orbit. At the poles and over the equator, the correction would be zero. The correction is also negligible for orbits larger than ~5000 km apoapse altitude. The corrections used in this test set were based on specific points in the MGS aerobrake mission, thus only one lookup variable, apoapse altitude, was required.

Each pass of the MGS data set could be readily examined prior to testing. This allowed predictions to be made regarding the expected performance of the algorithm. This was done by fitting a normal curve to the raw acceleration data.

A representative drag pass from this test is shown in Figure 3. Shown in the figure are the acceleration channels, the sequence predicted periapse time, and the PTE computed estimate of periapse time.

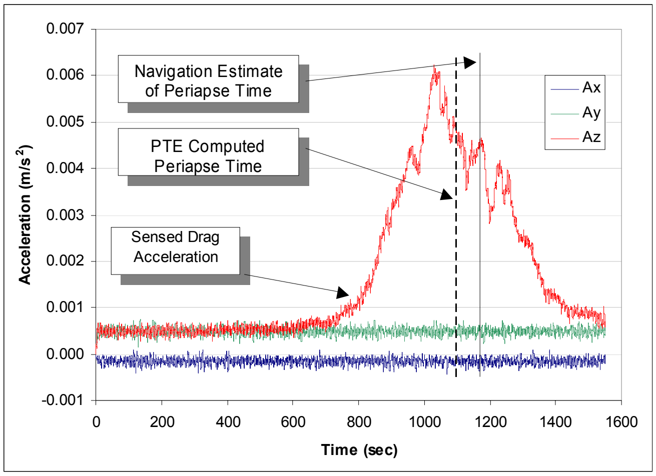

Figure 3 – Sample MGS Drag Pass Tested With PTE Algorithm

The test results are shown in Figure 4. The data shown is the error between the PTE estimate of periapse time and the navigation reconstructions of periapse time. The chart shows that the PTE algorithm was able to calculate the time of periapse within 80 seconds for all of the MGS cases tested.

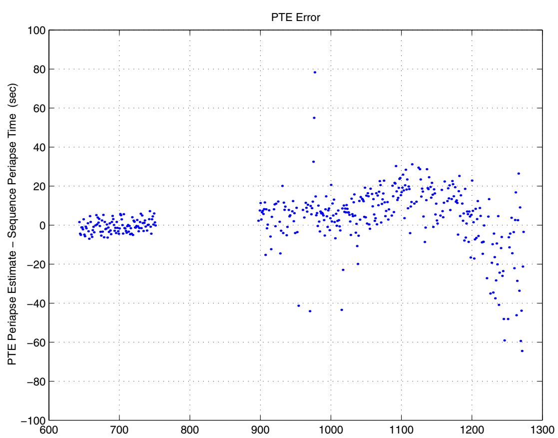

Figure 4 – PTE Algorithm Performance Relative to Nav Predicted Periapse Times

Since the PTE software has the ability to alter the execution time of the command sequence, it was necessary to ensure that it only did so under nominal conditions. Thus, a set of checks was developed to determine if an off-nominal condition existed. These checks included: verifying that the sequence periapse time is reasonable; ensuring that sufficient data has been collected; ensuring that the calculated drag delta-V is reasonable; verifying that the spacecraft is in the proper attitude control mode of operation; and verifying that the IMU is producing valid data. In the interest of testing time, these tests were performed on the actual flight code in a simplified environment.

Additional testing was performed to see how PTE interacted with the actual flight software in a flight-like environment. These tests required a significant amount of time and resources to execute, therefore only a limited set of behaviors could be tested. Typical tests of aerobraking operation only involve a few (e.g. 5 or less) short-period drag passes, and under normal conditions, a time shift large enough to be worth correcting should not occur. The solution was to adjust the spacecraft's true orbit such that the orbit period was slightly different from that assumed in the onboard sequence. An orbit period of 1.9667 hours, or 118 minutes was chosen, while the sequence assumed an orbit period of 2.00 hours, or 120 minutes. This resulted in a 2 minute timing error that increased with each orbit. The specific values were chosen such that the timing error would trigger a correction within the duration of the test.

The tests were set up such that the true periapse time and the sequence estimate of periapse time were in synch at drag pass one. The PTE algorithm was activated just prior to drag pass three, thus a 4 minute timing mismatch should have been introduced. The algorithm was allowed to correct the drift after that drag pass, but then was deactivated for all subsequent orbits.

The atmosphere used in the simulation was a simple exponential model, which was a limitation of the simulation at the time of testing.

Verification of proper algorithm performance involved 4 steps. The first check was to determine that the software did not think an off-nominal condition (as described earlier) existed and that the calculated time correction was actually provided to the on-board sequence. Next was to check the telemetry for the PTE algorithm and verify that a time correction of approximately –240 seconds was calculated. Third, comparing drag pass sequence execution time to a test without PTE active shows that the commanding was shifted in time. The final check is to ensure that the reference attitude profile was time-shifted as well.

The status of the PTE software from one of the verification tests is shown in Table 1, below. It shows that the algorithm was functioning normally and that 6437 data points were used in the calculations.

Table 1 – PTE Status Telemetry

Orbit	Time Tag	PTE Status	PTE Valid Data Points
1	01/341-16:15:01.199	N/A	N/A
2	01/341-20:02:00.102	NO ERROR	N/A
3	01/341-21:07:52.117	NO ERROR	6437

The determination of the magnitude of the commanded shift, as well as the verification that it was actually used to shift the next drag pass sequence can be seen in Table 2.

Table 2 – Periapse Times and PTE Performance

Orbit	Time Tag	Spacecraft Clock Time	Original Nav Predicted Periapse Time	Actual Stored Predicted Periapse Time	PTE Time Correction
1	01/341-16:15:01.2	N/A	N/A	N/A	Not active
2	01/341-20:02:00.1	N/A	18:39:18	18:39:18	Not active
3	01/341-21:07:52.3	692226485	20:39:23	20:39:23	-236.8 sec
4	N/A	N/A	22:39:25	22:35:28	Not active

The data shows that after the third drag pass in the test, a correction of ~237 seconds was calculated – only 3 seconds off from the error built into the test. Proper application of the time shift can be seen by comparing the fourth and fifth columns of the table. They show that the calculated time correction was used to shift the execution of the following drag pass sequence.

The last test of algorithm performance is examination of spacecraft attitude data. Nominally, the spacecraft is commanded to track the orbital velocity vector as it flies through a drag pass. This attitude profile is referenced to a specific time. A timing error in the execution of this tracking command will show up as an error in the spacecraft pitch angle, as shown in the attitude error plot in Figure 5. The X-axis attitude error corresponds to pitch. Assuming negligible yaw and roll, the pitch rotation is roughly in the orbital plane. Any angular error can then be converted into an equivalent timing error. For a 2 hour orbit, a two minute time error is equal to a 6 degree attitude error.

Drag pass one is not shown, but occurs at approximately 16:39. At drag pass two (the first one shown), the X-axis attitude error averages approximately 0.1 radians (5.7 degrees). This corresponds to roughly the 2 minute error that is expected due to the reduced orbit period. At the third drag pass (the first pass PTE is operational), the attitude error averages near 0.23 radians (13.2 deg). This translates to a timing error of 4.4 minutes, again roughly what was expected (4 minutes). At the fourth drag pass, the X-axis attitude error averages about 0.12 radians (6.9 degrees). This is equal to a 2.3 minute timing error, roughly equal to the error at drag pass two. Since the timing error rate is fairly constant (2 minutes/orbit), this means that the PTE algorithm did indeed apply a time correction of approximately 4 minutes between drag passes three and four. This is consistent with the test telemetry and calculations outlined above. Based on the differences between the pass two and pass four errors (0.3 minutes), the periapse time would have been corrected to within approximately 18 seconds.

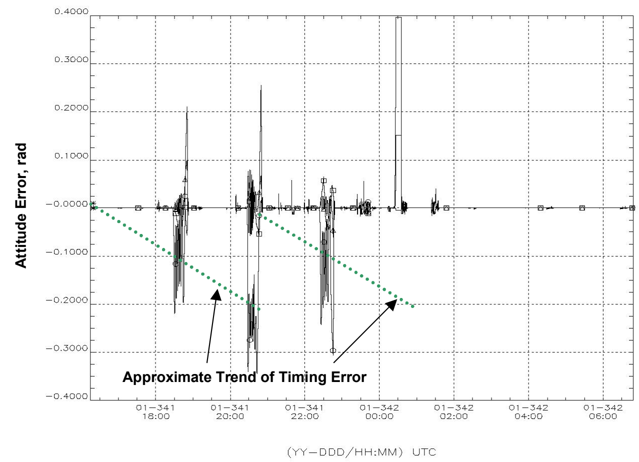

Figure 5 – Spacecraft Attitude Errors for Test Drag Passes 2, 3, and 4

3. ODYSSEY FLIGHT PERFORMANCE

Mars Odyssey was successfully injected into a 18.6 hr orbit of Mars on October 23, 2001 via a single burn of its main engine. Aerobraking was initiated three days later with a Walkin burn performed on October 26 that placed the orbital periapsis at 157 km. For the first half of aerobraking, PTE was operated in a background mode in order to verify its functioning since this region was relatively benign for timing errors. This was because the orbit was large enough that the ground navigation was able to reconstruct each orbit and accurately solve for the next one with minimal delay. In these large period orbits, small variations in drag pass delta-V result in large changes in orbit period, and correspondingly large changes in periapsis timing. Thus, the operations team generated a new Odyssey command sequence for every orbit when the orbit period was larger than 6 hours. This approach discovered and corrected timing errors prior to the each new drag pass. Uplink of a new command sequence cleared any adjustments computed by the PTE software. When the orbit period shrank to less than 6 hrs (more than 2/3 of the total aerobraking orbits were shorter than this), the period reduction for a given delta-v was small enough that the inherent PTE lag was acceptable. PTE's usefulness became more apparent with these shorter orbits.

On December 19, 2001 it was decided to enable the PTE software in its active mode. In this middle phase of aerobraking, the timing performance was of intermediate value (Figure 6). In this phase of the mission, navigation predicted 3 orbit sequences with 2 actually being used onboard due to processing and uplink delays. The inherent navigation timing error was up to 150 seconds with PTE limiting this error to 100 seconds. Around December 31, PTE was enabled for 4 orbit predictions and showed very good performance, 300 second navigation timing errors were routinely reduced to 100 seconds or less (Figure 7). In the final phase of aerobraking, the dynamic pressures were reduced to a level where the inherent timing errors were relatively small and PTE's usefulness was reduced. Figure 8 shows the navigation timing error was less than 150 seconds with the PTE controlling the resulting sequence error to about 100 seconds. All of the PTE timing control updates successfully maintained the orbiter's aerobraking sequence within its base requirements.

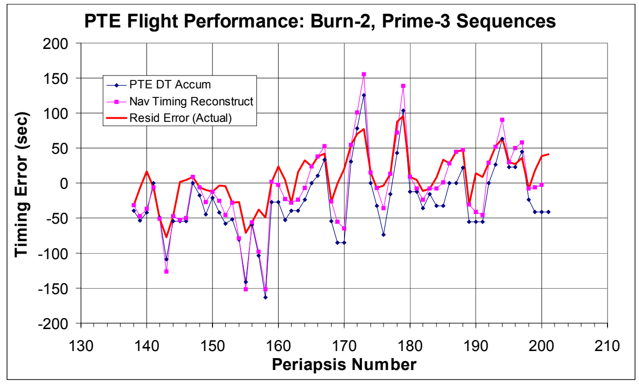

Figure 6 – PTE Performance Phase I

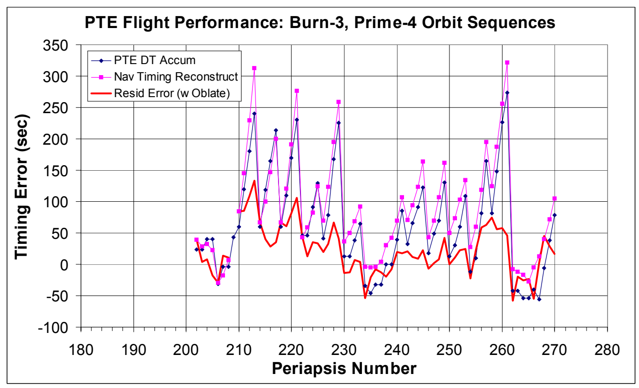

Figure 7 – PTE Performance Phase II

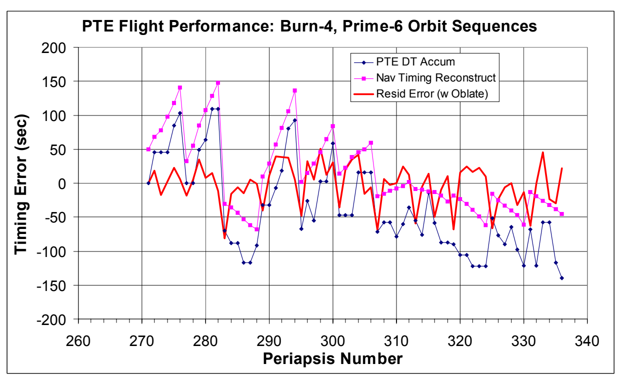

Figure 8 – PTE Performance Phase III

The residual PTE timing errors for all of Odyssey aerobraking are shown in Figure 9.

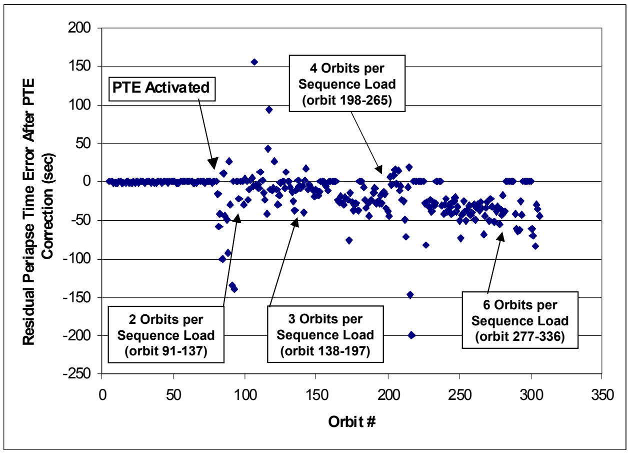

Figure 9 - PTE Performance During Odyssey Aerobraking

4. CONCLUSIONS

The Future – Following Odyssey's Footsteps

The next Mars orbiter, the Mars Reconnaissance Orbiter (MRO), is currently being developed. Like Odyssey before it, the prime contractor for the MRO spacecraft is Lockheed Martin. This provides the opportunity to reuse and improve the software used on previous missions, including Mars Odyssey, Mars Climate Orbiter, and Mars Global Surveyor. MRO will also utilize aerobraking to transition from a 35 hour post-MOI orbit into a sub-2 hour science orbit over the course of approximately 6 months.

The PTE software has moved beyond demonstration status and has been baselined as an integral part of the flight software. Due to the larger capture orbit of MRO, more time will be spent in long-period orbits, the evolution of which are more sensitive to the variations in drag pass delta-V. To deal with this, the PTE software will be enhanced to provide more accurate predictions of future periapse times by comparing the amount of delta-V measured by the spacecraft with the amount predicted by the navigation team. This should allow PTE to be activated earlier, improve accuracy, and allow more time for the operations team to build and test the products used on the spacecraft.

Additional enhancements will allow the PTE software to determine whether any heating constraints have been violated in a drag pass, and signal the need to raise the altitude of upcoming drag passes. These improvements will help make aerobraking operations safer and more robust.

Final Thoughts

The Periapse Time Estimator software was successfully demonstrated during Mars Odyssey aerobraking operations as a useful means of increasing spacecraft autonomy. In the initial phase of the demonstration, it was observed that the PTE timing calculations did not compare well to the reconstructed errors observed by the navigation team. This was expected and understood - since the algorithm was optimized for middle to late aerobraking it lacked the ability to incorporate drag delta-V into the timing corrections. This capability is necessary for accurate operation in large-period orbits. Once the orbit periods had fallen below 6 hours, real-time PTE performance matched navigation reconstructs very well, thus the decision to allow the timing corrections to be applied to the operating command sequence. Having PTE functioning onboard provided significant relief to the operations team, as they were no longer pressured to make manual corrections themselves.

Acknowledgements

The authors would like to recognize the hard work and dedication from everyone involved with the Mars 2001 Odyssey Program at Lockheed Martin Astronautics and the Jet Propulsion Laboratory. Their efforts have made this mission an example for all others to follow. We would also like to specifically recognize the efforts of Geoffrey Hauser at Lockheed Martin Astronautics, who was responsible for coding the flight-version of the PTE algorithm and for generating the tests that helped evaluate the algorithm's interaction with the rest of the Odyssey flight software.

5. REFERENCES

[1] W. Willcockson, "Magellan Aerobraking Control Corridor - Design & Implementation," *AIAA/AAS Space Flight Mechanics Conference,* AAS 94-117, February 1994.

[2] J. Beerer, R. Brooks, P. Esposito, D Lyons, W. Sidney, H. Curtis, and W. Willcockson, "Aerobraking at Mars: the MGS Mission," *AIAA 34th Aerospace Sciences Meeting,* Reno, NV, AIAA 96-0334, January 15-18, 1996.